SURROGATE MODELS

FOR STRUCTURAL ESTIMATION AND UNCERTAINTY QUANTIFICATION

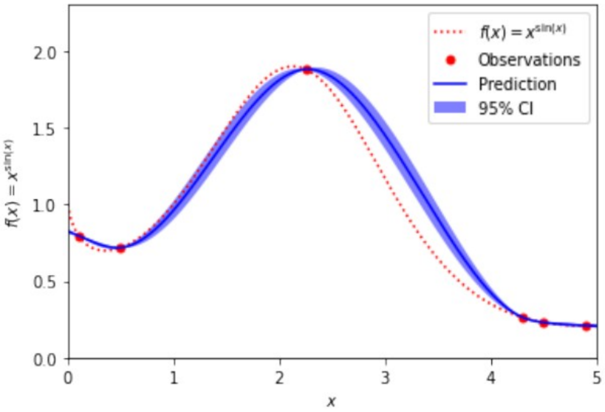
University of Geneva

March 25th, 2025

https://github.com/sischei/Deep_Learning_Geneva_2025

Simon Scheidegger

simon.scheidegger@unil.ch



UNIL | Université de Lausanne

Roadmap of this lecture

I. GP surrogates
   I. Bayesian active learning
   II. Surrogates for Structural Estimation
   III. Surrogates in Option Pricing

Bayesian Active Learning

- With the recent explosion of available data, you have can have millions of examples with a high cost to obtain labels.
- For instance, when trying to predict the sentiment of tweets, obtaining a training set can require immense manual labour.
- But worry not, active learning comes to the rescue!
- In general, active learning is a framework allowing you to increase classification performance by intelligently querying you to label the most informative instances.

Reinforcement Learning

-Computing globally accurate optimal policies is a challenging task.

- Thus far, we have placed the observation points to train the Gaussian processes randomly inside the relevant part of the state space (simplex)

- This strategy can be highly inefficient

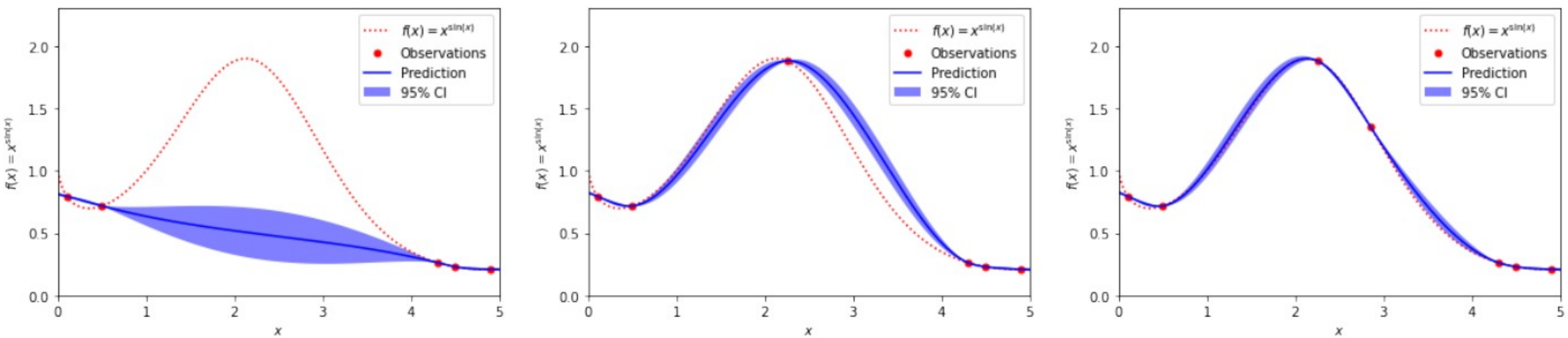
→ **Bayesian Active Learning** (see, e.g., Deisenroth et al. (2009))

- technique from the reinforcement learning to automatically place observations in regions of the state space where they improve most

  • on the quality of the approximator. → day4/code/BAL_with_GPs.ipynb

- Score Function

$$U(\tilde{x}) = \sigma_m \mathbb{E}\left[V^\tau(\tilde{x})|\mathbf{X}\right] + \frac{\sigma_v}{2}\log\left(\text{var}\left[V^\tau(\tilde{x})|\mathbf{X}\right]\right)$$



Bayesian Active Learning

- see day4/code/active_learning_gpytorch.ipynb

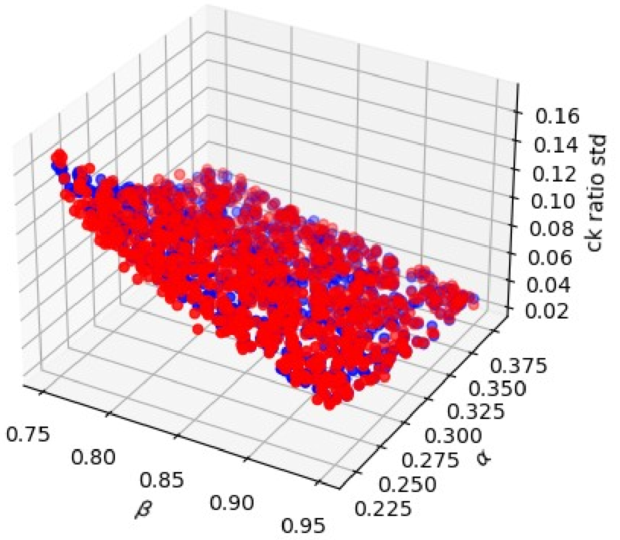
DEQN surrogate combined to GPs

- Typically, we are not only interested in the policies as a function of states and parameters, but moments.
  → Solutions can be used to generate to simulate moments, and other quantities of interest.

- Example: fit a Gaussian Process to the consumption-capital ratio and try to find parameter combinations that match the long run average and standard deviation.

- Let’s have a look at a stochastic growth-model with parameters as pseudo-states.
  Code: **Deep_Learning_For_Dynamic_Econ/lectures/day4/code/DEQN_production_code**

- Run model: python run_deepnet.py
- Simulate model: two steps – first export path of results, then simulate.
  $ export USE_CONFIG_FROM_RUN_DIR= home/PATH_TO_YOUR_SOLUTIONS
  - python3 post_process_GP.py STARTING_POINT=LATEST hydra.run.dir=$USE_CONFIG_FROM_RUN_DIR



Pricing options with GPs (BS surrogate model)

- see day4/code/GP-BS-Pricing_01.ipynb

Greeks

- see day4/code/GP-BS-Pricing_02.ipynb

- The GP provides analytic derivatives with respect to the input variables

$$\partial_{X_*}\mathbb{E}[\mathbf{f}_*|X, Y, X_*] = \partial_{X_*}\boldsymbol{\mu}_{X_*} + \partial_{X_*}K_{X_*,X}\alpha$$

$$\partial_{X_*}K_{X_*,X} \;=\; \frac{1}{\ell^2}(X - X_*)K_{X_*,X}$$

$$\alpha \;=\; [K_{X,X} + \sigma_n^2 I]^{-1}\mathbf{y}$$

- Second-order sensitivities → diff. wrt. X $*$

Summary on GPs

- For a fairly simple idea, Gaussian processes do tend to work very well on a wide range of topics.
- The way that the covariance function explicitly encodes the correlations that can be seen in the data means that the user has a lot of control.
- Even in the simple treatment here we have put quite a lot of effort into making the computations numerically stable and relatively fast.
- However, there is much more that can be done, including methods for approximation to speed things up significantly.